ISPRS Journal of Photogrammetry and Remote Sensing 191 (2022) 302–314

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Connectivity-aware Graph: A planar topology for 3D building surface reconstruction☆

Shengming Yang [a], Guorong Cai [a,*], Jing Du [a], Ping Chen [b], Jinhe Su [a], Yundong Wu [a], Zongyue Wang [a], Jonathan Li [c]

[a] School of Computer Engineering, Jimei University, Xiamen 361021, China
[b] The Departments of Information and Communication Engineering, North University of China, Taiyuan 030051, China
[c] Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada

ARTICLE INFO

Keywords:
3D reconstruction
Building surface reconstruction
Mesh processing
Simplification
Connectivity graph

ABSTRACT

Multi-view Stereo (MVS) meshes suffer from occlusions or missing data, making most surface reconstruction methods invalid. Due to data imperfections, existing methods, just like PolyFit, have made a trade-off between reconstruction accuracy and time consumption. We propose a novel approach that automatically reconstructs a building surface model from raw triangular mesh with data incompleteness. Unlike existing methods that extract high-quality primitives, our method focuses on assemblies of primitives to control the level of geometric detail in the reconstructed models. We design a topological relationship of the primitives to form a plane connection graph. To be precise, we first combine the scalability of shapes with plane primitives. Then strong and soft connections between primitives are constructed by calculating the confidence of the plane intersection in space. Furthermore, the topological relations of all primitives are encoded into an undirected graph. Finally, a watertight and manifold model is extracted from the faces of a candidate set by energy minimization. Experiments on the Helsinki 3D dataset demonstrate the superiority of our method in time consumption and reconstruction error, as measured by Hausdorff distance. Our method outperforms other primitive-based algorithms in handling non-planar structures. Even when dealing with imperfect data, a watertight model is still obtained.

1. Introduction

Reconstructing buildings in 3D scenes is a challenge for the photogrammetry and remote sensing community. In recent years, requirements for surface reconstruction of point clouds and Multi-View Stereo (MVS) meshes in industry and academia have continued to expand so that models can indeed be applied to real scene applications, such as using building models for digital cities and virtual reality.

The main challenge for reconstruction is to obtain high-precision, lightweight surface models, taking into account the problem of imperfect data (e.g., large storage, noise, and deficiency due to occlusion). Reconstruction methods are divided into two main categories.

High precision. The first goal of reconstruction is high precision. Deep Neural Networks (DNNs) learn complex and sharp surface details to reconstruct high-precision surfaces of small and medium-sized objects. However, the main problem for DNNs is the computational cost required for large scenes (Mi et al., 2020; Luo et al., 2021).

Time consumption. Secondly, researchers take time consumption as the standard. State-of-the-art methods reconstruct the surface of the overall contour of an object in a large scene by analyzing a series of a priori topological relations. Nevertheless, obtaining a higher accuracy surface model with a tolerable time consumption is remaining a research hotspot (Li et al., 2016; Nan and Wonka, 2017; Bouzas et al., 2020).

Most current reconstruction methods demand input data of high quality, because noisy and missing data dramatically impact reconstruction results. The motivation of our work is to assure the reconstruction pipeline is compatible with flawed (occlusion and sparse) data. Our idea is to complement a model on the basis of plane hypothesis,

☆ This work was supported in part by the National Natural Science Foundation of China under grant no. 41971424 and no. 61902330, the Key Technical Project of Xiamen science and technology bureau, China under grant nos. 3502Z20191018, 3502Z20201007, 3502Z20191022, 3502Z20203057, the Science and Technology Project of Education Department of Fujian province, China under grant nos. JAT190321, JAT190315, the Natural Science Foundation of Fujian Province, China under Grant 2020J01701, 2022J01337, 2022J01819, the Key Project of the Natural Science Foundation of Fujian Province, China under Grant 2022J02045.

* Corresponding author.
E-mail address: guorongcai.jmu@gmail.com (G. Cai).

https://doi.org/10.1016/j.isprsjprs.2022.07.024
Received 21 December 2021; Received in revised form 19 June 2022; Accepted 26 July 2022

i.e., assume the model consists of only planar primitives, while keeping time consumption within acceptable limits by graph data structure. Meanwhile, starting from existing prior knowledge, pre-processing, or attaching hard constraints to the cost function (Nan and Wonka, 2017) becomes an effective strategy. This tactic is the foundation of our approach.

Based on the plane hypothesizing strategy in PolyFit (Nan and Wonka, 2017), we propose a new topological relationship of spatial planes to generate models using different positional relationships of planes in space, which improves the surface accuracy of the output model, and, to a certain extent, solves the problem of defects due to occlusion. Urban meshes covering about 4 km^2 in Helsinki (Finland) are the input for the algorithm (Gao et al., 2021). The goal is to reconstruct a watertight model even when a part of a building is missing due to occlusions of surroundings.

In order for the reconstruction method to tackle well the problem of missing occlusion in sparse data, we formulated several goals that must be achieved in a reconstruction pipeline: (1) When the algorithm is applied to less noisy, less missing data, the reconstruction method must capture effectively geometric details, such as walls and corners. (2) In the case of imperfect data as input, a reconstructed model still describes the overall contour of the building and has the properties of a watertight, two-dimensional manifold. (3) The time consumption of the proposed method outperforms an exhaustive segmentation-based method, such as PolyFit (Nan and Wonka, 2017). Besides, under-detection of primitives significantly impacts existing methods. To reduce the impact of this problem on a plane-based method, the proposed algorithm uses soft connection vertices. Therefore, because the lack of planar primitives has less impact on our approach, we focus mainly on generating a watertight surface model using detected planes. The significant contributions of our work are as follows:

- A lightweight data structure and a general strategy are designed to address the problem of data defects in a building for various reasons, such as tree occlusion.
- A confidence strategy combining graph structure is proposed, which, compared with PolyFit, reduces time consumption. In addition, our method generates LoD2 models that capture more accurate geometric details while reducing errors.

2. Related work

Surface reconstruction issues involve computer graphics and computer vision. The review of previous literature covers four aspects directly related to our proposed method: (a) Extract high quality primitives to generate an accurate model. (b) Partition space and extract the model. (c) Assemble primitives to form a closed surface. (d) Approximate surface, relying on rough shape.

Planar shape detection. Compared with natural objects, most artificial objects, such as buildings, comprise many planes. Detecting the plane primitives is to divide the initial input point set into multiple sets of interior points, with the points in each set representing a plane they fit. At present, random sample consensus (Schnabel et al., 2007) and region growing (Marshall et al., 2001; Rabbani et al., 2006) are the most mature methods for initial primitive detection that still obtain reliable detection results when the input data is noisy and defective. In addition, shape primitives are extracted by learning approaches trained from CAD model databases (Fang et al., 2018; Li et al., 2019). All these methods perform well in experiments; however, it is always problematic to determine the initial parameters, which produce over- and under-detection of planar primitives (Chin et al., 2018). Therefore, to reduce the impact of this problem, our algorithm focuses on establishing an assembly relationship for existing detected primitives.

Partitioning-based reconstruction Partitioning-based methods divide 3D space into polyhedral spaces by extending planar primitives (Verdie et al., 2015; Boulch et al., 2014; Oesau et al., 2014). An output model is formed by selecting a subset of candidates' faces. The major problems are the tremendous consumption of time and memory space. Recently, researchers have focused on decreasing computational costs. PolyFit (Nan and Wonka, 2017) filters and simplifies detected primitives to obtain the final model from a candidate face set based on binary linear programming formulation. A kinetic data structure is proposed to divide space into smaller polyhedral spaces and then obtain surface models by solving a graph-cut problem (Bauchet, 2019; Bauchet and Lafarge, 2020). Furthermore, Connect-and-Slice proposes a method of combining planar primitive connectivity with slicing (Fang and Lafarge, 2020). By not exhaustively slicing the space and incorporating the scalability of the shapes into the algorithm, our approach significantly reduces time consumption.

Connectivity-based reconstruction. Connectivity-based methods attempt to construct adjacency relationships between planar primitives (Schindler et al., 2011; Chen and Chen, 2008). Researchers focus on constructing different topological relations of planar primitives in space and combine the connection graph with three-plane intersections (Chen and Chen, 2008; Schindler et al., 2011; van Kreveld et al., 2011). When input data is missing or there is over-detection of the primitives, using this strategy is in vain. An adjacency graph emerges with wrong adjacency relationships, thereby invalidating the result. One way to overcome this problem is an interactive solution (Arikan et al., 2013). Structure-Aware Building Mesh Polygonization (SABMP) (Bouzas et al., 2020) makes the adjacency condition more stringent, resulting in a more straightforward connection graph. Because our approach designs a new planar relationship and encodes different relationships into an undirected graph, the algorithm is more robust to imperfect data.

Mesh simplification. To meet the needs of industry and animation, many studies have focused on the simplification of 3D meshes. First, a surface, approximated from a raw triangle mesh, is simplified. Therefore, researchers, initially focused on reducing the storage space of the mesh designed a cost function to constrain the simplification process of the mesh. Quadric Error Metrics (QEM) (Garland and Heckbert, 1997) was proposed as a simplification based on quadratic errors as the cost. Variational Shape Approximation (VSA) (Cohen-Steiner et al., 2004) constructs planar proxies based on an input model through cost function constraints. When the storage requirement is satisfied, researchers consider the speed and accuracy of the algorithm; thus, the introduction of spatial topological relations is also a necessary step. To achieve the goal of simplifying the overall model, Structure-Aware Mesh Decimation (SAMD) (Salinas et al., 2015) attempts to simplify the detected planar proxies. To simplify the building mesh, SABMP (Bouzas et al., 2020) encodes the topological relationship of the initial planes into an undirected graph. Therefore, our proposed method uses prior knowledge to analyze topological relationships in space to improve time efficiency and model accuracy.

3. Method

As seen in Fig. 1, our method consists of the following three processing stages: (1) Primitives detection and build-up of the strong-connected graph (see Section 3.1). (2) Search for primitives that satisfy the soft-connected condition (see the red wire-frame) and encode relationships into the strong-connected graph (see Section 3.2). (3) Surface extraction using an energy minimization formulation (see Section 3.3).

3.1. Primitives detection and strong connected graph

Our method uses a raw triangular mesh as input. Using the region growing method for city models (Lafarge and Mallet, 2012), we first extract a set of primitives. Furthermore, the algorithm captures the intrinsic topological relationship between primitives from the original model.

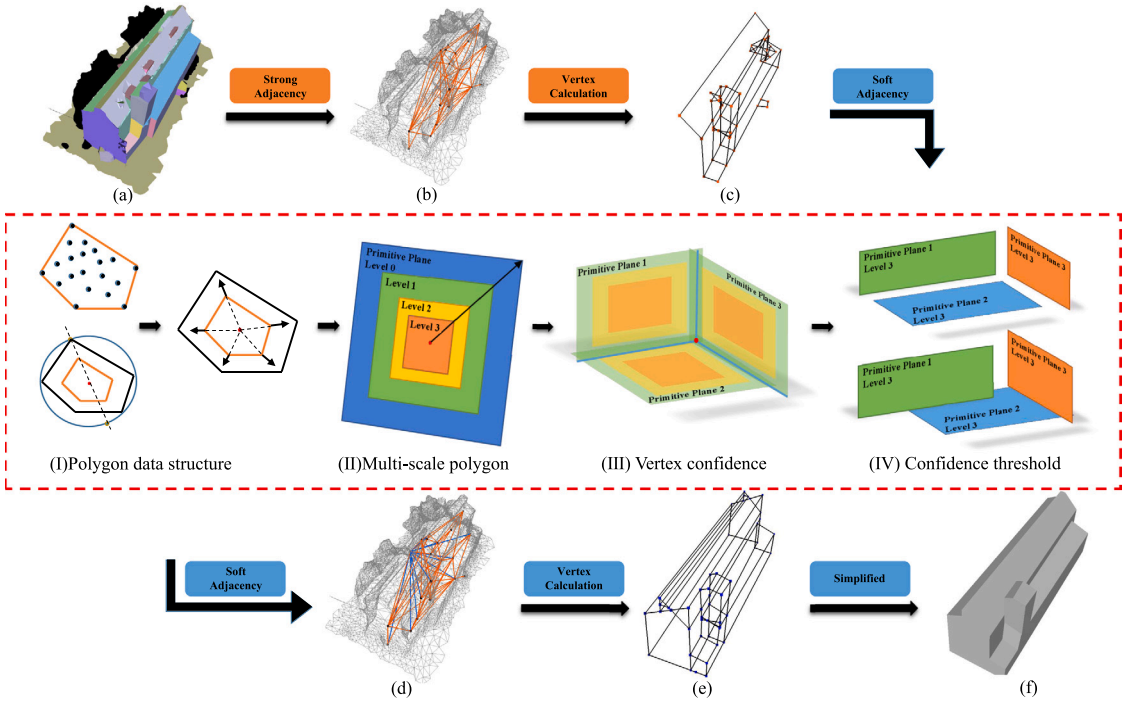

Fig. 1. Pipeline of the proposed 3D model reconstruction method. Our algorithm starts from a mesh with a set of primitives represented by colored polygon (a). By calculating the strong connection between the primitives (see orange edges), we build the strong-connected graph (b). The building scaffold (see the black wire-frame) is generated using only the strong-connected graph (c). Through the calculation of the soft connection relationship (see the red dashed box), the algorithm adds soft connections based on the strong-connected graph (see blue edges) in order to form the soft-connected graph (d). The building scaffold is generated based on the soft-connected graph (e). The output model with watertight and 2D-manifold characteristics (f). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

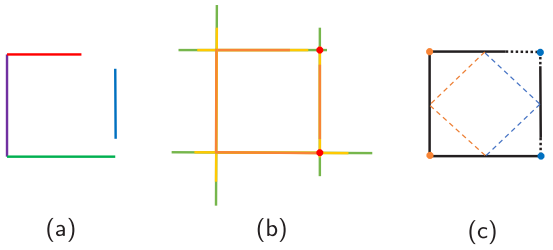

Fig. 2. A simple 2D illustration of a soft connected graph. (a) Primitives detection(each primitive is represented by a separate color); (b) Plane growth(three colors of each primitive represent three growth scales; red dots represent the intersection of the primitive growth); (c) Connection graph(orange dots and dashed lines represent strong connection nodes and strong connection relationships; blue dots and dashed lines represent soft connection nodes and soft connection relationships). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

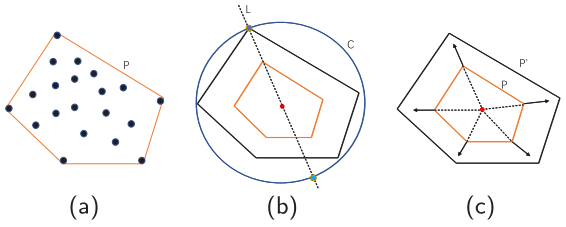

Fig. 3. Plane shape growing data structure. (a) Minimum convex polygon; (b) Circumscribed circle for one point(orange polygon represents the base polygon in Fig. 3(a)); (c) Plane shape growing(black polygon represents the polygon from which the base polygon in Fig. 3(a) has grown). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Existing methods have different ways of generating a strong connected graph (Bouzas et al., 2020; Fang and Lafarge, 2020). However, the notion of strong connectivity is defined as characterizing the relationship between two primitives that are adequately spatially close. The case where the interior points of two planar primitives share at least one edge is defined as a strong connection relationship.

The algorithm encodes the interrelationships detected between primitives to an undirected graph. Each graph vertex represents a primitive. There are two types of graph edges: strong-connected edges and soft-connected edges.

3.2. Soft connected graph

In our method, only the adjacency relationship of different primitives in a small spatial neighborhood is regarded as a strong connection. It is difficult to capture adjacency relationships from noisy and missing data; therefore, a soft connection relationship is designed to solve this problem. As shown in Fig. 2, we provide a 2D toy example to describe the soft connection idea. In terms of algorithm implementation, first, based on the plane hypothesizing strategy, the easiest way to deal with missing data is to extend the plane to achieve model completion. Next, because the excessive extension of the plane causes more consumption of memory and time, a hyperparameter determines the degree of plane growth. Finally, to speed up the reconstruction solution, a confidence strategy is employed to associate plane growth with the connectivity graph.

3.2.1. Plane shape growing

Inspired by Kinetic Shape Reconstruction (KSR) (Bauchet and Lafarge, 2020), we developed an algorithm that designs a new structure to grow planar shapes. The difference is that we consider the concept of scale only (see Fig. 3).

The scalability of shapes in our method is expressed through the lightweight data structure defined by the following three points:

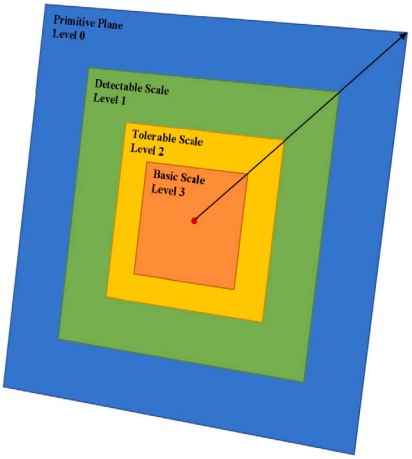

Fig. 4. Confidence of multi-scale polygon.

(a) For the set of interior points of each primitive, the algorithm projects the shape to the fitting plane, maps the 3D points to 2D points, and calculates the smallest convex polygon P for the 2D point set on the plane.
(b) Each vertex of the convex polygon P has the following two attributes: the center of gravity on the straight-line L formed by the vertex, and the circle C passing through the vertex with the barycenter as the center.
(c) The growth of the convex polygon P' is achieved by controlling the radius of each circle to grow proportionally to the length of its initial radius P.

3.2.2. Confidence of multi-scale

We anticipate that the algorithm perceives the scalability of a shape. Therefore, the expansion of planar convex polygons is defined according to three scales(basic, tolerable expansion and detectable), each of which is assigned a different confidence level.

The confidence level of a convex polygon decreases as its area increases. The polygon with the smallest area has the highest confidence in the basic scale; whereas, the polygon with the largest area has the lowest confidence in the detectable scale. It is seen that the ratio of the area between polygons with different confidence levels becomes the key to measuring the expansion and contraction of a plane primitive. The algorithm defines the missing scale coefficient k of the initial model, which is a user-defined percent parameter between 0 and 1. Three growth parameters correspond to the basic, tolerable and detectable scales. In fact, we use the missing scale coefficient k to ensure that the growth of the three scale polygons is at the same scale. In Section 4 (Experiments), the scales of the three levels are defined as $1.1^t, 1.4^t, 2^t$, where $t = \tan 0.65\pi k$. We set the highest confidence to basic scale, the lowest to detectable scale, and the confidence of the plane primitive to 0 (see Fig. 4). The confidence of a point projected onto a plane is equal to the confidence of the convex polygon with the smallest area. For example, when a point is in a polygon with a tolerable scale, but not in the polygon with a basic scale, its confidence is level 2.

3.2.3. Soft connected vertex

After calculating a multi-scale convex polygon for each planar primitive, the algorithm computes the intersection points of all planar primitives and adds them to the point set S. Each intersection point is generated from the intersection of three planar primitives. If the intersection point is outside the 3D bounding box of the initial model, it will be discarded.

When the point set S is obtained, the confidence level of each intersection point will be calculated. Each intersection is formed by three planes (see Fig. 5). The intersection points are projected onto three planes. The confidence of the intersection point is the sum of the confidence of the three projection points. Actually, this step of the method expresses the topological relationship of the three planar primitives in space through the confidence level of the intersection. The intersection points with higher confidence represent the closer spatial relationship of the three planar primitives in real space.

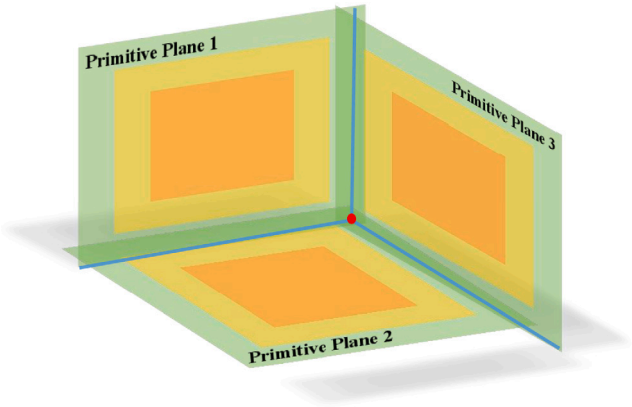

Fig. 5. Intersection of three adjacent primitives.

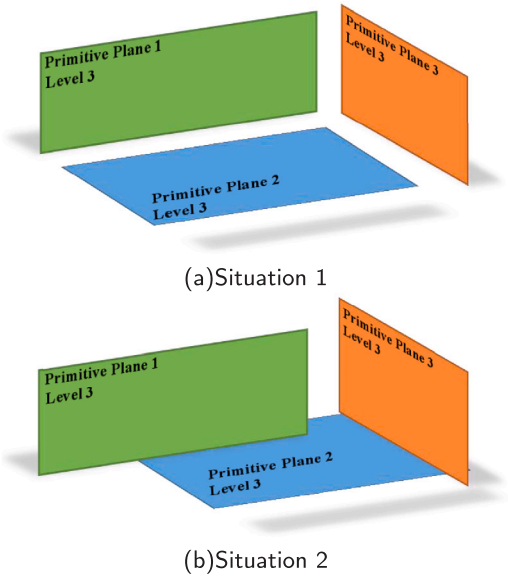

Fig. 6. Two situations of triple planes' intersection.

(a) In Fig. 6(a), when the three planes cannot form a strong connection relationship, the relationship in the space of the basic scale polygon (the smallest convex polygon) formed by the three planar primitives is assumed. Suppose two projection points fall within a convex polygon with a tolerable scale, and one projection point falls into a polygon of the detectable scale. In that case, we consider that these three planes form a soft connection at this time.
(b) In Fig. 6(b), a pair of planes in the three planes forms a strong connection relationship. If two projection points fall within the convex polygon of the detectable scale and one projection point falls within the basic scale polygon, then we consider that these three planes form a soft connection.

We use the topological relationship between the above two planes in space as a threshold to prove whether an intersection is reliable or

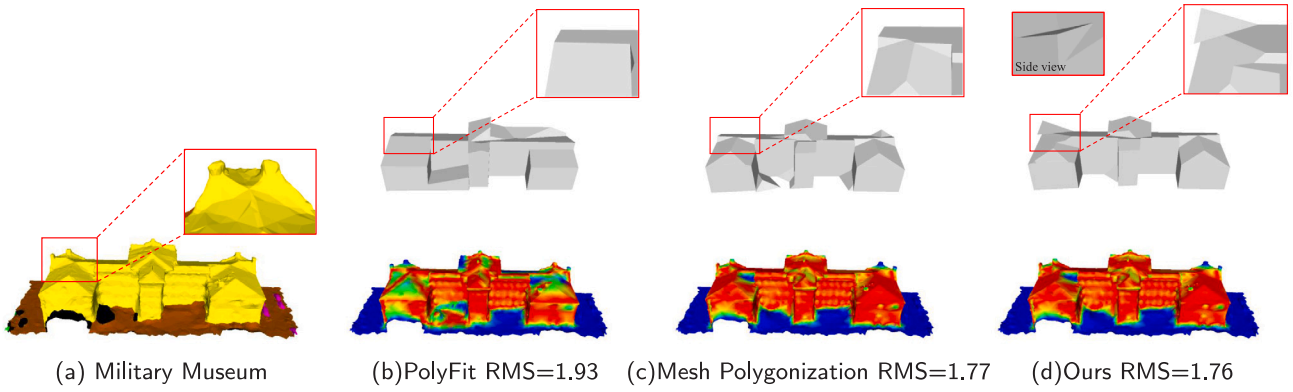

Fig. 7. Reconstruction result of *Military Museum*.

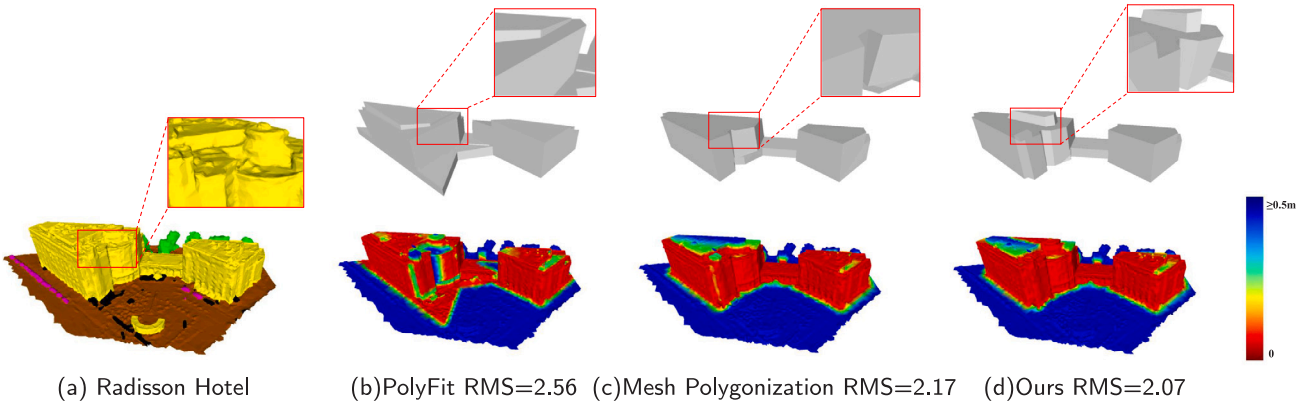

Fig. 8. Reconstruction result of *Radisson Hotel*.

not. The confidence of the intersection in the above two cases is 5. If the confidence of the intersection is equal to or greater than 5, the relationship will be judged as a soft connection. The maximum and minimum confidence of intersections are 9 and 0, respectively. The above cases list only the two threshold cases in the algorithm to judge whether an intersection is reliable or not.

The soft connection relationships between the planes are encoded into a planar graph, and all strong connection relationships satisfy the conditions of the soft connection. Therefore, to form a soft-connected graph, we add soft connection edges based on the strong-connected graph.

3.3. Surface extraction

The generation of the final model is based on the formulation of a binary linear programming problem (Papadimitriou and Steiglitz, 1982; Williams, 2009). To generate the set of candidate faces, we improved the strategy of forming a building scaffold in Mesh Polygonization (Bouzas et al., 2020).

The building scaffold is generated from a soft-connected graph, where each simple cycle formed by three primitives in the soft-connected graph represents a vertex. If there is a soft connection between the plane primitives, the vertex is defined as a soft connection node. Otherwise, the vertex is a strong connection node. The candidate face set is generated by mapping through the planar primitives to the building scaffold.

The indicator variable x_i represents whether the candidate face f_i is selected ($x_i = 1$) or not ($x_i = 0$). The indicator variables are connected by minimizing the energy function under the constraints of manifold and watertightness. The energy term of an indicator variable is derived from the development of the energy term of Mesh Polygonization (Bouzas et al., 2020), which includes four aspects: face coverage, data fitting, model complexity, and defect completion term.

Face coverage. The face coverage term is defined as based on the area of the candidate face covered by the α-shape (Edelsbrunner et al., 1983) formed by the initial triangular face.

$$E_c = \frac{1}{A(bbox)} \sum_{i=1}^{N} x_i \cdot (A(f_i) - A(M_i^a)) \tag{1}$$

where $A(bbox)$ is the surface area of the bounding box of the input model, $A(f_i)$ is the area of the candidate face f_i, and $A(M_i^a)$ is the face area covered by the α-shape formed by the initial model.

Data fitting. The data fitting term is related to the number of triangular faces covering the candidate face.

$$E_f = 1 - \frac{1}{|F|} \sum_{i=1}^{N} x_i \cdot s(f_i) \tag{2}$$

where $|F|$ is the total number of faces in the initial model and $s(f_i)$ is the number of faces covering the candidate face.

Model complexity. Model complexity term is related to the number of corners and creases in the final model. This term is set to avoid gaps and enforce large planar regions.

$$E_m = \frac{1}{|E|} \sum_{i=1}^{|E|} c(e_i) \tag{3}$$

where $|E|$ is the total number of edges in the final model. $c(e_i)$ is an indicator function. When the selected adjacent candidate faces are coplanar, the value is 0, otherwise 1.

Defect completion. The setting of the defect completion term is related to the vertices of the candidate face generated by the soft connection. Soft connections represent close spatial relationships that

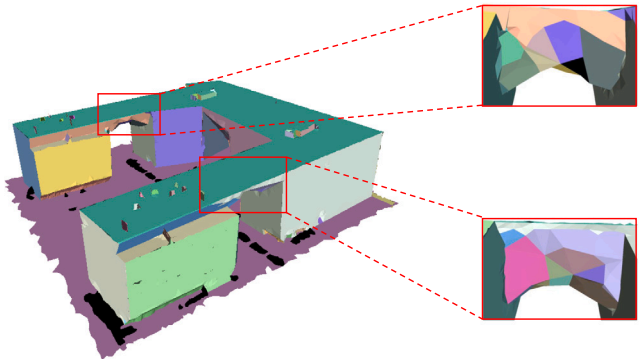

Fig. 9. Large curvature structure in the input model.

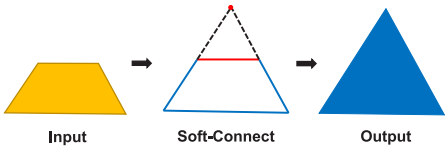

Fig. 10. 2D example of soft connection.

are not sufficiently detected by the initial model. The defect completion item is more inclined to select candidate faces formed by soft connections. The premise of this term is based on the assumption that the initial input model is defective.

$$E_d = 2k(1 - \frac{1}{|C|}\sum_{i=1}^{|C|} x_i \cdot v(f_i)) \tag{4}$$

where $|C|$ is the total number of candidate faces. k is the missing scale coefficient, $v(f_i)$ is the weight of the vertex of candidate face f_i.

$$v(f_i) = \frac{1}{|V|}\sum_{i=1}^{|V|} w(v_i) \tag{5}$$

where $|V|$ is the number of vertices of candidate face f_i; $w(v_i)$ is the vertex weight. If the vertex on the candidate faces is a soft connected vertex, then $w(v_i)$ is equal to 1, otherwise 0.

By minimizing the weighted sum of the energy as mentioned in the above terms, a subset of the candidate face set is selected to generate the final simplified model. Meanwhile, strong constraints are required to ensure that the generated model has manifold and watertight characteristics (Nan and Wonka, 2017), as given by Eq. (6).

$$s.t. \begin{cases} \min_x \lambda_f E_f + \lambda_c E_c + \lambda_m E_m + \lambda_d E_d \\ \sum_{j \in N(e_i)} x_j = 2 \text{ or } 0, \quad 1 \le i \le |E| \\ x_i \in \{0,1\}, \quad 1 \le i \le N \end{cases} \tag{6}$$

In the case where the initial input model has defects, the optimized model calculated by the IP solver (Williams, 2009) still has an overall profile.

4. Experiments

Dataset. We applied our method to the SUM Helsinki 3D dataset (Gao et al., 2021), whose annotations are divided into six categories: Ground, Vegetation, Building, Water, Vehicle, and Boat. To evaluate our method, throughout the experiments, we collected 126 buildings of various architectural styles.

The experimental data are divided into two categories: intact and defective. The intact data are used to compare our algorithm with existing advanced technology. The challenging data with flaws prove the robustness of the method to missing data.

Implementation details. We implemented our method in C++ using the CGAL library (Alliez and Fabri, 2016), which provides the basic geometric tools for mesh-data structures. The primitive detection of all methods in an experiment uses the same standard methods from the CGAL library. Also, all methods are optimized using the same mathematical solver. The optimizer used in solving the final model is SCIP solver (Vigerske and Gleixner, 2018). For the parameter settings, $\lambda_f = 0.39$, $\lambda_c = 0.24$, $\lambda_m = 0.27$, $\lambda_d = 0.1$. The percentage of user input determines the missing scale coefficient. We designed that, when k is greater than 0.5, the initial data is considered not reliable enough. In an experiment, there is no input greater than 0.5; therefore, the defect completion term E_d has a scale factor of 2, which expands the influence of k on the optimization.

4.1. Data without occlusion

To compare our methods with state-of-the-art methods, we chose (1) PolyFit, which performs surface reconstruction by infinite partitioning of space, and (2) Mesh Polygonization, which reconstructs a surface using only the strong connectivity relations we have defined. These two methods correspond to the two extreme cases of the algorithm, that is the cases where the missing scale coefficients are 1 and 0, respectively.

Fig. 7 through Fig. 14 show the reconstruction results of six different architectural styles under different methods. The Hausdorff distance between two meshes (Guthe et al., 2005) is used as an indicator to measure the difference between the reconstruction results of different methods and the original model.

The building, *Military Museum*, (see Fig. 7(a)) is simple in overall structure, but with geometric details. Our reconstruction results portray the overall contours and attempt to generate (albeit not accurate enough) geometric details (see Fig. 7(d)) that are missing in the other models. This is also a point that the Mesh Polygonization reconstruction results do not have (see Fig. 7(c)). Due to uneven ground, the small plane formed on the ground significantly impacts the hypothetical process of PolyFit, and the plane formed on the ground divides the overall building and affects the partition result in Fig. 7(b).

Fig. 8 shows a building with a connecting corridor bridge. It is a challenge for existing methods to reconstruct separated buildings while retaining geometric details. The plane intersecting and plane confidence of multiple buildings affect the reconstructed result of PolyFit, and generate a less complex model (see Fig. 8(b)). The advantage of our method is that an undetectable small plane can be abstracted into a point by soft connection, maintaining geometric accuracy (see Fig. 8(d)). However, in Mesh Polygonization, the number of planes detected remains stable as the surface area of the input mesh increases (see Fig. 8(c)).

Figs. 11 and 12 show two buildings with grooves and sharp geometric details. Comparison results of the two groups fully illustrate the effectiveness of setting the missing scale coefficient. In Fig. 11(b) and Fig. 12(b), PolyFit describes the overall structure; however, unnecessary plane intersections produce wrong results, which is the main problem of slice-based methods. Mesh Polygonization looks for effective closed loops in graph structures formed only by strong connectivity relationships, and it is easy to lose the overall structure (e.g., groove façade and cylindrical surface) (see Fig. 11(c) and Fig. 12(c)). When the missing scale factor in Fig. 12(d) is adjusted, our results balance the relationship between geometric details and the whole contour. However, as shown in Fig. 11(d), there are still slight deviations.

Fig. 13 shows a building with sharp details with large curvature at the joints and a simple overall structure. Moreover, we aim to use building, *Aallonkoti Hotel* (see Fig. 13(a)), to illustrate why our proposed method works well.

The plane detection algorithm has two basic parameters: the minimum fitting number of a single primitive and the tolerance distance of the plane interior point. When a user reduces the two parameters, more planes are detected. The disadvantage is that, when the surface

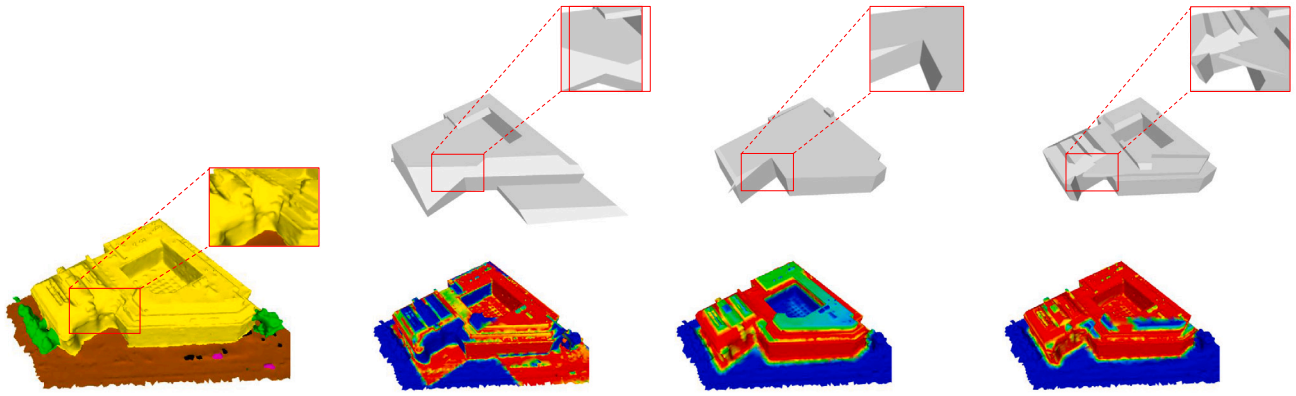

Fig. 11. Reconstruction result of *Aiko Mall*.

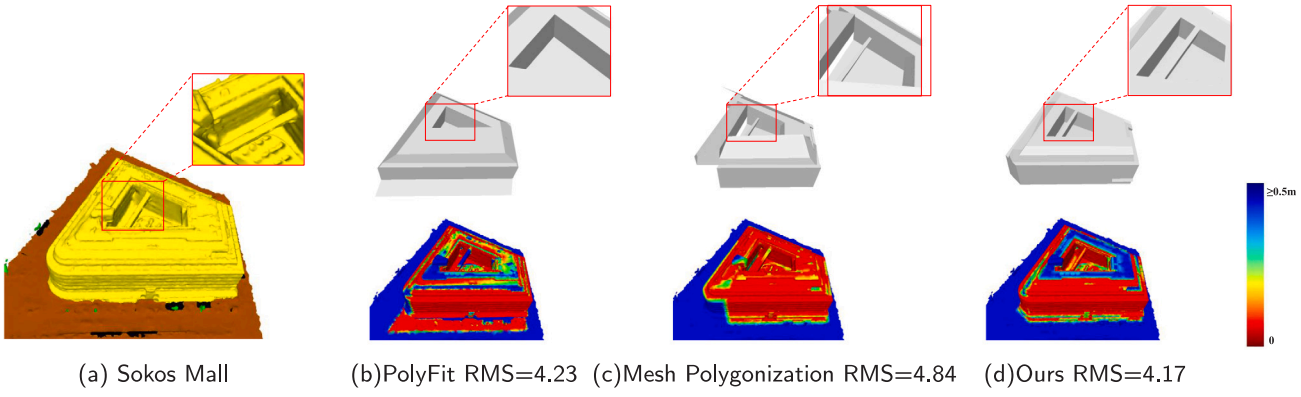

Fig. 12. Reconstruction result of *Sokos Mall*.

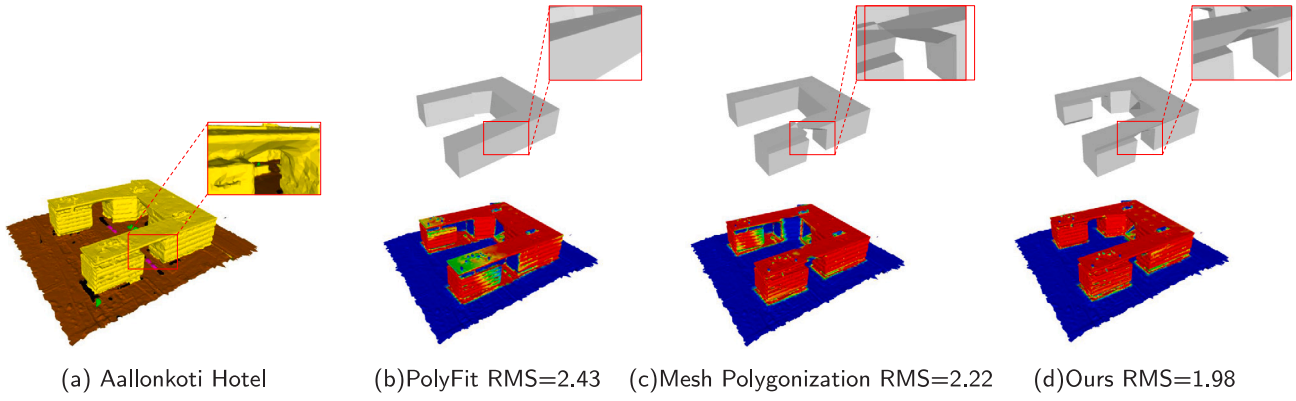

Fig. 13. Reconstruction result of *Aallonkoti Hotel*.

Table 1
Quantitative evaluation for models presented on Figs. 7 to 14.

Building			PolyFit (Nan and Wonka, 2017)				Mesh Polygonization (Bouzas et al., 2020)				Ours			
	Points (original)	Faces (original)	Primitives	Points	Faces	Simplify (%)	Primitives	Points	Faces	Simplify (%)	Primitives	Points	Faces	Simplify (%)
Military Museum	6384	12447	14	347	340	2.7	85	151	262	2.1	85	292	580	4.6
Radisson Hotel	17414	34148	28	389	408	1.2	131	89	174	0.5	185	416	766	2.2
Aiko Mall	28460	56525	21	441	446	0.7	223	75	146	0.2	381	528	944	1.7
Sokos Mall	23927	47484	22	294	300	0.6	182	98	200	0.4	292	422	844	1.8
Aallonkoti Hotel	18050	35825	31	210	216	0.6	111	85	166	0.5	178	567	1130	3.1
Kluuvi Mall	27023	53577	28	592	640	1.2	161	113	222	0.4	244	717	1248	2.3

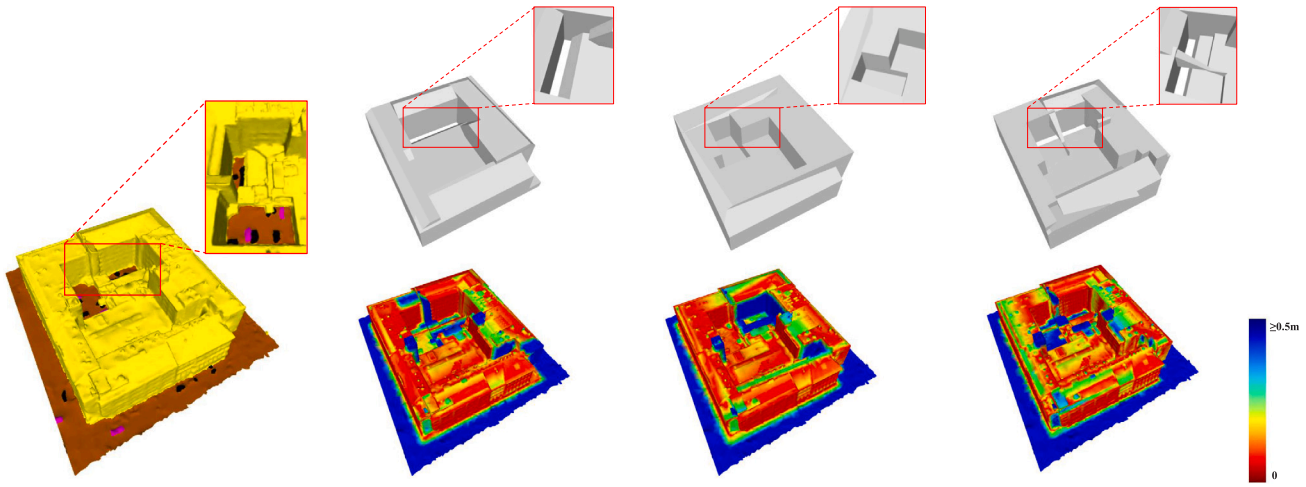

Fig. 14. Reconstruction result of *Kluuvi Mall*.

Table 2
Quantitative evaluation for models presented on Figs. 7 to 14.

Method	*Military Museum*		*Radisson Hotel*		*Aiko Mall*		*Sokos Mall*		*Aallonkoti Hotel*		*Kluuvi Mall*	
	Time (s)	RMS	Time (s)	RMS	Time (s)	RMS	Time (s)	RMS	Time (s)	RMS	Time (s)	RMS
PolyFit (Nan and Wonka, 2017)	2	1.93	249	2.56	431	6.61	321	4.23	181	2.43	232	3.57
Mesh Polygonization (Bouzas et al., 2020)	9	1.77	26	2.17	53	3.89	40	4.84	38	2.22	71	3.31
Ours	9.5	1.76	82	2.07	156	3.28	84	4.17	68	1.98	196	3.41

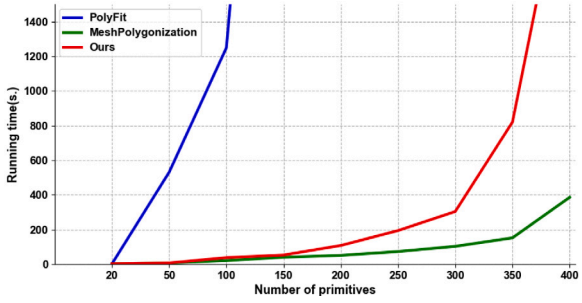

Fig. 15. Performances of surface reconstruction methods.

curvature of the input model is large, the model consists of more plane primitives (see Fig. 9).

Each of the three algorithms removes a part of the planes by specifying the minimum fitting number of plane primitives to reduce the influence of noise. Nevertheless, when users want to capture geometric details with larger curvatures in the model, more planes must be added to the calculation. Both our algorithm and Mesh Polygonization introduce the concept of adjacency graphs. However, if strong connections only are used, when the curvature is large, it is likely that an effective closed-loop cannot be formed in the adjacency graph. Furthermore, a strong closed constraint is used in the process of energy function optimization. The emergence of this situation causes a part of the output model to be missing (see results in Fig. 11(c) and Fig. 12(c)). Our soft connection solves this problem well:

Fig. 10 is a two-dimensional example to describe how a soft connection handles the under-detection problem. The blue border line represents the primitive selected by the algorithm. The red line represents the planes whose interior points are less than the minimum threshold or the missing data caused by occlusion. When only the strong adjacency relationship is used, the entire model cannot satisfy the closed constraint; therefore, a watertight model cannot be generated.

To generate a closed watertight model, our method calculates the confidence of the red vertices and establishes soft connections for the blue border lines. The function of the soft connection is apparent as the algorithm abstracts the plane (red line), which has small-scale interior points, into a point (red vertex) having the same adjacency relationship with the plane. Alternatively, it can be viewed as a way to complement missing data, thereby generating a geometric model with more detail.

The above is why our method deals well with the under-detection of the data of primitives. In any case, the algorithm abstracts the undetected planes into points (see Fig. 13). The number of early detection planes also determines the LODs level of the generated model.

Fig. 14 shows a challenging building to test the performance of the algorithm based on planar primitives when a large number of planes is detected. PolyFit is very sensitive to the number of plane primitives. The time efficiency of PolyFit increases exponentially with an increase in the number of primitives. Thus, PolyFit abstracts only a rough outline (see Fig. 14(b)). When our method deals with a large number of planes, the number of soft connections increases exponentially with the number of primitives, but optimization time is reduced. Although the reconstruction result is not as good as the result of Mesh Polygonization in terms of Hausdorff distance (see Fig. 14(c)), it is seen that our method generates as many more geometric details as possible with fewer errors (see Fig. 14(d)).

Table 1 shows the number of input planar primitives in different methods. Based on the same time efficiency, our method handles a larger number of input planes than the other two methods. PolyFit, which relies significantly on a greedy slicing strategy, has an enormous computational time cost. Mesh polygonization is fast and lightweight, but is extremely limited by the quality of the input data. It can be seen from the number of points and faces of the reconstructed model that our algorithm obtains a high-precision model with accurate geometric details, such as the building, *Aallonkoti Hotel*. The number of obtained points and faces of the output model is two to three times that of other

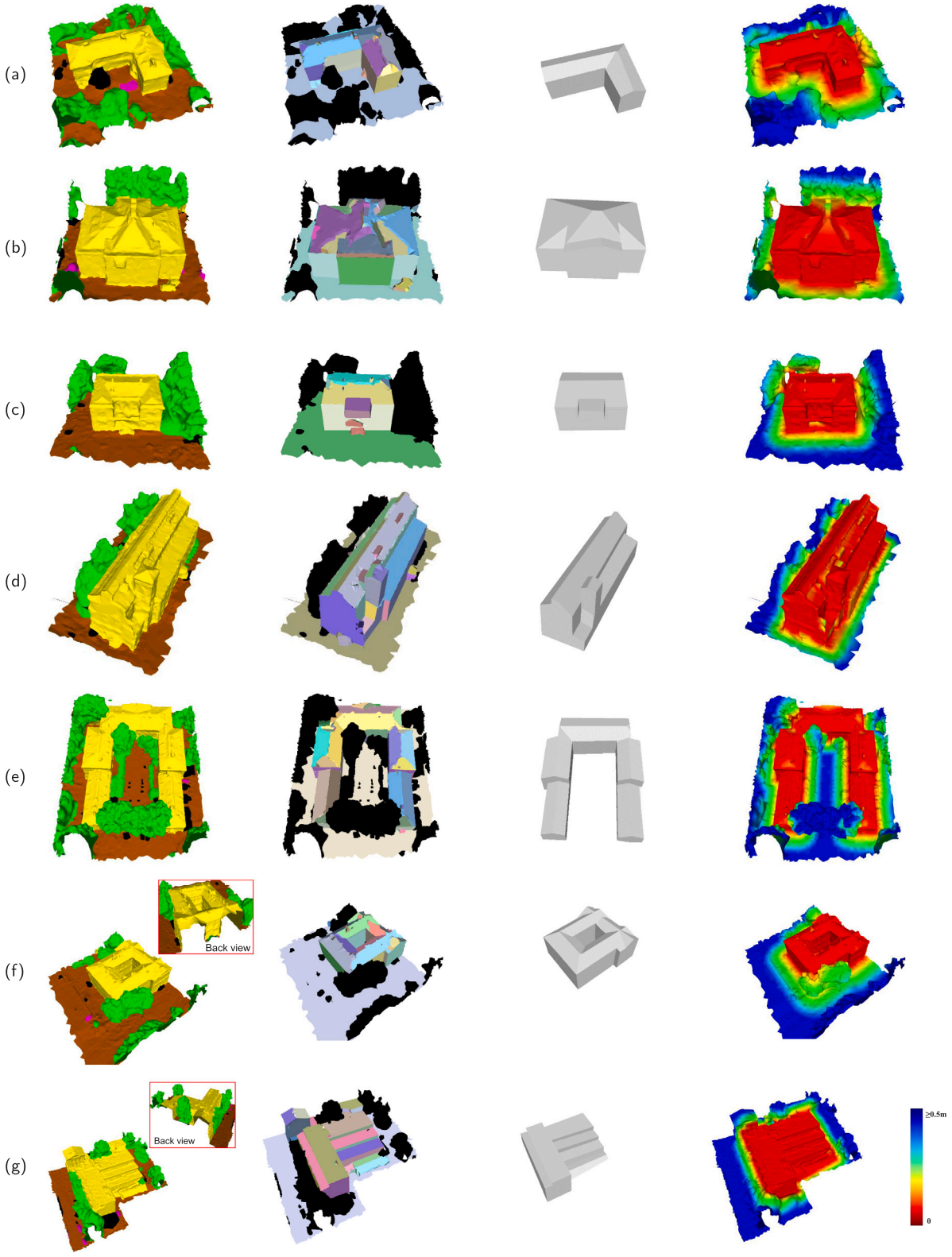

Fig. 16. Reconstruction results. From left to right: original model, detected primitives, reconstruction model, the visualization of the Hausdorff distance defined between the input model and the result.

methods, and the error is smaller. The simplified ratio represents the storage space ratio of the result to the input model. Because our method is designed for retaining numerous geometric details, the ratio of our reconstruction results, stable at about two percent, is slightly higher than that of other algorithms.

Timings.

Table 2 shows the time efficiency and root-mean-square error, in different methods, calculated by Hausdorff distance. The limitation of PolyFit is that the solution time for the energy function increases exponentially with an increase in the number of initial plane primitive

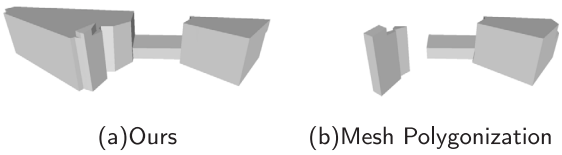

Fig. 17. Reconstruction of the *Radisson Hotel* when the number of primitives is consistent.

Table 3
Comparison between our method with PolyFit and Mesh Polygonization in Fig. 17.

Method	Primitives	Strong connection	Soft connection	Time
PolyFit (Nan and Wonka, 2017)	155	–	–	≥2 h
Mesh Polygonization (Bouzas et al., 2020)	155	79	–	18 s
Ours	155	79	90	55 s

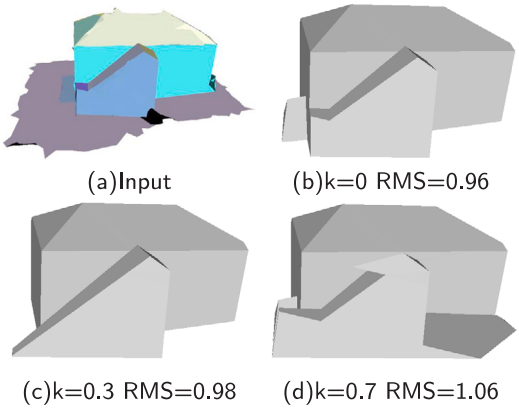

Fig. 18. Reconstruction of the building by gradually increasing the ratio of the missing scale coefficient.

inputs. Fig. 15 shows the relationship between the calculation time of the three algorithms and the number of initial detection primitives when the fixed missing scale coefficient is 0.3. It is seen from Fig. 15 that the function of the soft connection we designed is to find an essential balance between the goal of complementing missing data or capturing more geometric details and time consumption. Theoretically, the time efficiency of our proposed method completes high-precision geometric detail reconstruction based on time efficiency infinitely close to Mesh Polygonization.

Table 3 shows the time efficiency of our method and the other two methods with the same number of input primitives. In comparison experiments, for the case of Mesh Polygonization (Bouzas et al., 2020) with the same number of input primitives as our method, the generated reconstruction results are often zero solutions due to the overly complex strong connectivity graph. In particular, Fig. 17 shows the reconstruction results. Note that the execution of PolyFit terminated after two hours, so we cannot visually compare the results of PolyFit (see Table 3 and reconstruction results). Compared with PolyFit and Mesh Polygonization, our method sacrifices a little time efficiency but improves the robustness of the accuracy to imperfect data.

4.2. Data with occlusion

Here, we consider only the semantically labeled meshes of ground and buildings as input for the following two reasons: (i) The mesh that ignores other semantic tags when used as algorithm input results in the input building simulation missing due to occlusion; (ii) To achieve more accurate and precise reconstruction, the algorithm is used as a post-processing step for the result of semantic segmentation.

Fig. 16 shows the results of our method applied to seven buildings, with varying degrees of data missing due to occlusion. The reconstruction difficulty of seven buildings with missing data ranges from low to high. And building f and g do not have closed characteristics. Our method does not use the hypothesis of PolyFit for building data with missing data, i.e., the plane of the 3D bounding box is added to generate easily a watertight model. Therefore, the planes detected by the input model must intersect in space to form at least a closed space, which can be seen from the reconstruction results of f and g.

Our proposed method, when used on missing data, cannot capture the precise details of a structure. However, Table 4 shows that the time efficiency is within an acceptable range when applied to models with missing data. The final models generated all have the properties of watertight and manifold. We still count the RMS error used to measure the difference between the reconstruction result and the original model. However, due to the influence of vegetation data in the initial model, the RMS error cannot fully measure the effectiveness of our algorithm. The reconstruction results of insufficient data meet only the visualization requirements.

Effect of the missing scale coefficient.

Given the value of the degree of missing scale coefficients, the algorithm generates results according to the specified parameters, which also means that our method is susceptible to changing a missing scale coefficient. Fig. 18 shows this process. When a missing scale coefficient is gradually increased, the algorithm adds more soft connection vertices to the final model. The whole process is considered an "association" process. To complete the model, the algorithm calculates according to the specified parameters. We still record the Hausdorff distance between the input and reconstructed results.It is seen that the error increases as the coefficient increases.

Ablation study.

Fig. 19 shows the effect of the face coverage and the data fitting terms under different coefficients. Note that the face covering term tends to choose planes with high confidence; whereas, the data fitting term expects all input data to be included in the computation of surface extraction. The ratio of the coefficients of the two terms balances the effect of noise on the reconstruction results. Fig. 19(b) shows that the data fitting term tends to have low noise in the input data; however, the reconstruction result produces some sharply wrong details due to noise. The face coverage term in Fig. 19(c) prefers the input be highly noisy. The reconstruction result contains only some high-confidence planes and loses the overall contour. The validity of the regular term weights used in the experiments has been verified in PolyFit (Nan and Wonka, 2017). Note that a wide range of weights produces the same results.

Fig. 20 shows the effect of the defect completion term. If the weights of the edges in the connectivity graph are identical, then our proposed method simply complicates the connectivity graph and increases the time consumption compared to Mesh Polygonization. The defect completion term is used to balance the weights of the edges with different attributes in the connectivity graph. Actually, soft connection has a higher rank in the connection graph, which means that the energy term is inclined to select intersections formed by soft connections. This makes the reconstruction method easier to complement the model. To illustrate this feature, we added the ablation experiment shown in Fig. 20. This preference also produces unnecessary errors, such as building *Aiko Mall*. Thus, the selection of energy term coefficients also minimizes the influence on the reconstruction results relative to other energy terms ($\lambda_d = 0.1$ in the experiment).

Although the algorithm requires the user to input only a missing scale coefficient to cope with the absence of the model, the parameters of the planar growth are still predefined. In fact, we found these parameters must be determined by the point density of the input data and the degree of data deficiency.

Table 4
Statistics on the result presented in Fig. 16.

Building	Points (original)	Faces (original)	Primitives	Time (s)	Points (simplified)	faces(simplified)	RMS
(a) *Botanic Garden*	6573	12607	38	2	33	62	1.08
(b) *Residence*	10113	19564	58	23	220	444	1.72
(c) *Museum*	14299	27920	64	27	175	346	3.03
(d) *Company*	6860	13173	59	4	43	82	2.17
(e) *Museum*	6995	13722	57	5	59	114	1.81
(f) *Restaurant*	23684	46460	74	18	225	446	2.44
(g) *Archives*	6351	12262	30	2	29	54	1.51

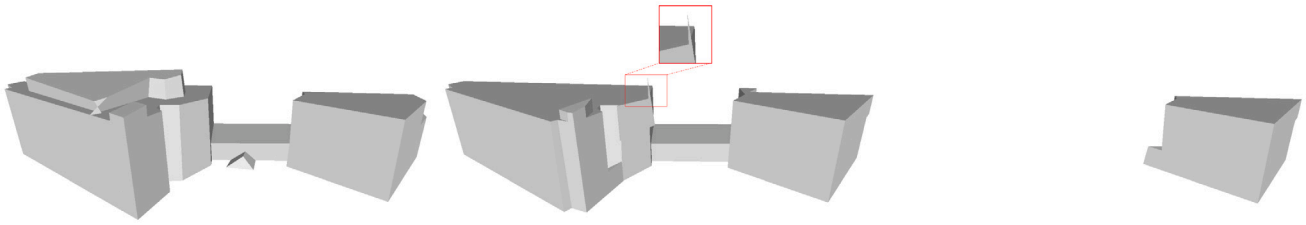

Fig. 19. Reconstruction results of *Radisson Hotel* under different energy coefficients.

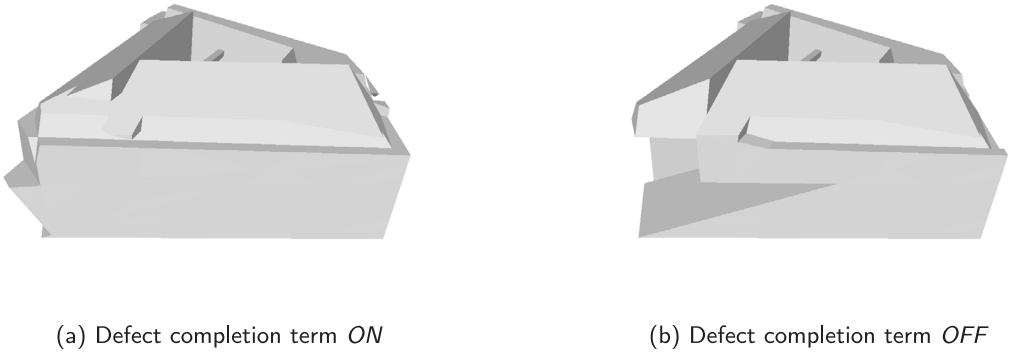

Fig. 20. The effect of the defect completion term (reconstruction results of *Sokos Mall*).

We then decided to set the values of the two parameters separately. The degree of missing data in the model is specified by the user. The basic parameters of planar growth are adjusted empirically for datasets with different point densities. To illustrate the principle of parameter setting, we tried multiple combinations of scale parameters for the current dataset (see results in Tables 5 and 6). As can be seen from the tables, the values of the scale parameters directly affect the number of soft connections formed. Moreover, the number of soft connections is related to the time efficiency of the computation. Note that the time efficiency of the algorithm is very sensitive to the choice of parameters.

In subsequent work, we will focus also on automatically determining parameters to improve the scalability of the proposed scheme.

The guidelines for parameter setting.

In order to obtain accurate reconstruction results, we provide a reasonable parameter setting guideline to adjust the missing scale coefficients. In Section 3.2.1, we calculate the convex polygons for planar shapes, which means that the calculation is inaccurate for buildings with non-convex polygons in the profile projection.

Fig. 21 shows a two-dimensional toy example for different parameters. In Fig. 21(a), the different colored line segments represent too small planes or missing data. Fig. 21(b) shows that the confidence level of the projected points is incorrectly increased when they fall into the blue region. In Fig. 21(c), when the missing coefficient is small, an under-segmented structure may be generated in the interior of the input polygon. Such an error is found in Fig. 22(c). When the missing coefficient is set larger, the generated over-segmented structure (see Fig. 21(e)) is outside the input polygon due to the incorrectly increased confidence of the projection points, just as the extended structure of the error in Fig. 22(e).

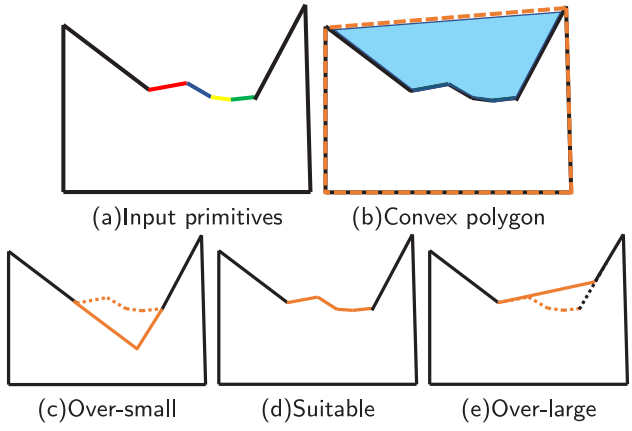

Fig. 21. Two-dimensional example of parameter setting. (a) Input plane primitives, different color line segments represent different planes; (b) Convex polygon, the orange dashed line represents the convex polygon of the plane shape, and the blue area represents the area of confidence increase error; (c) Over-small parameters; (d) Suitable parameters; (e) Over-large parameters.

Table 5
Statistics on the result presented in Fig. 22.

Scale parameters	(1.0, 1.1, 1.3)	(1.0, 1.2, 1.6)	(1.0, 1.3, 1.8)	(1.1, 1.4, 2.0)*	(1.1, 1.5, 2.2)
Soft connection	40	75	96	124	136
RMS	1.91	1.99	1.97	1.73	1.82
Time	14 s	73 s	101 s	152 s	206 s

Table 6
Statistics on the result presented in Fig. 22.

Scale parameters	(1.1, 1.7, 2.4)	(1.2, 1.4, 1.8)	(1.2, 1.6, 2.4)	(1.2, 1.7, 3.0)	(1.2, 2.0, 3.5)
Soft connection	163	118	157	210	280
RMS	1.86	1.96	2.04	1.97	2.03
Time	354 s	110 s	410 s	1056 s	≥2000 s

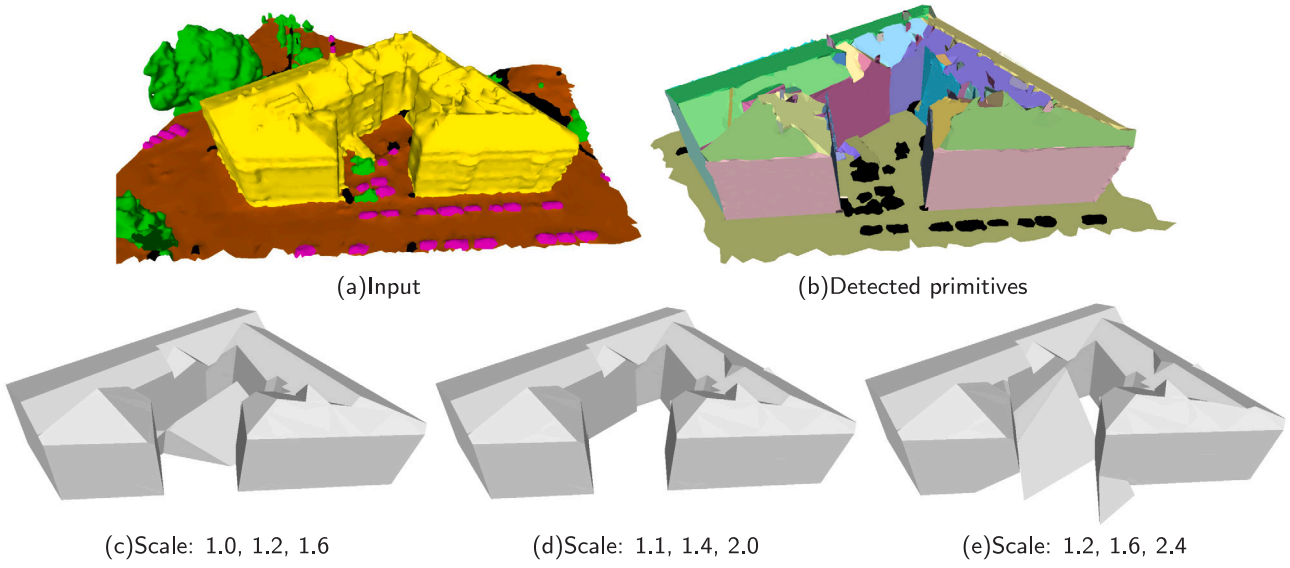

Fig. 22. The effect of the plane scale parameters (k=0.3).

As shown in Fig. 22, the basic growth parameters in Section 3.2.2 are reliable for the current dataset, while avoiding the case of inability to form an effective closed-loop in the connection graph (Fig. 22(c)) and plane overgrowth (Fig. 22(e)).

In summary, we can adjust the parameters reasonably by the position of the error structure generated by the reconstruction results according to the following guidelines:

(1) The reconstructed results are initially obtained with the basic parameters set on the dataset.
(2) By determining which of the generated errors corresponds to the case in Fig. 21, the parameter size is adjusted directionally.
(3) The accurate reconstruction results are obtained by iterating through step (2).

It is worth mentioning that for buildings with convex polygons in the building profile projection, the missing scale coefficient has a wide range to obtain the same results. The difference is only the computation time (see results in Tables 5 and 6). Our future work will focus on addressing the issues shown in Fig. 21.

Limitations.

At the same time, our simulation experiments tested the effect of our method under extreme pressure. As shown in Fig. 23(a), the number of points of the model is halved as input. It is seen that, although the missing data is filled in the results, errors are also generated (see Fig. 23(b)). The reason is that because the input to the algorithm has only the missing scale coefficient, it cannot determine which part of the data should be calculated and completed. The two planes in Fig. 23(c) (pointed to by the red arrow) are considered to be complemented after calculation. Generating this inaccurate structure is a major limitation in the proposed method. This topology error is also captured on the roof of Building *Museum(c)*.

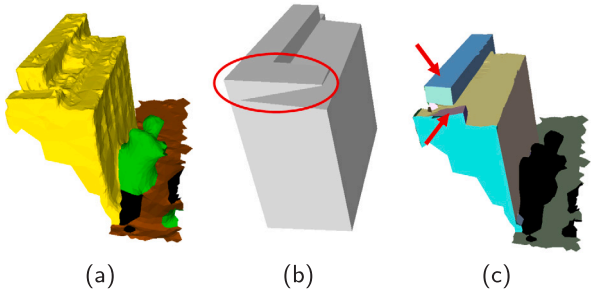

Fig. 23. Reconstruction under the limit of data defect.(a) Simulation model with flaws; (b) Reconstruction model; (c) Detected primitives.

5. Conclusion

We proposed a novel building surface reconstruction method based on the establishment of different connection relationships between plane primitives. Our proposed method, based on the strong and soft connection relationship we defined, rectifies the defect of not being able to generate watertight models when a certain degree of data is missing. The key lies in the soft connection relationship composed of

four data structures. The experimental results fully demonstrate the potential of the method in terms of time efficiency and accuracy.

The missing scale coefficient, which dramatically impacts the accuracy and time efficiency of the algorithm, also has certain limitations. In future work, we aim to show that the algorithm automatically determines the parameters. Improving the robustness of the algorithm to noise will also be the focus of future work. By application, we anticipate that our method can be combined with semantic segmentation, which can be used as a post-processing step for semantic segmentation of large scene meshes for surface reconstruction.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Alliez, P., Fabri, A., 2016. CGAL: the computational geometry algorithms library. In: Special Interest Group on Computer Graphics and Interactive Techniques Conference. SIGGRAPH '16, Anaheim, CA, USA, July 24-28, 2016,Courses, ACM, pp. 8:1–8:8. http://dx.doi.org/10.1145/2897826.2927362.

Arikan, M., Schwärzler, M., Flöry, S., Wimmer, M., Maierhofer, S., 2013. O-snap: Optimization-based snapping for modeling architecture. ACM Trans. Graph. 32 (1), 6:1–6:15. http://dx.doi.org/10.1145/2421636.2421642.

Bauchet, J., 2019. Kinetic data structures for the geometric modeling of urban environments. (Structures de données cinétiques pour la modélisation géométrique d'environnements urbains). (Ph.D. thesis). University of Côte d'Azur, Inria, Nice, France, URL https://tel.archives-ouvertes.fr/tel-02432386.

Bauchet, J., Lafarge, F., 2020. Kinetic shape reconstruction. ACM Trans. Graph. 39 (5), 156:1–156:14. http://dx.doi.org/10.1145/3376918.

Boulch, A., de La Gorce, M., Marlet, R., 2014. Piecewise-planar 3D reconstruction with edge and corner regularization. 33, (5), pp. 55–64. http://dx.doi.org/10.1111/cgf.12431,

Bouzas, V., Ledoux, H., Nan, L., 2020. Structure-aware building mesh polygonization. ISPRS J. Photogramm. Remote Sens. 167, 432–442.

Chen, J., Chen, B., 2008. Architectural modeling from sparsely scanned range data. Int. J. Comput. Vis. 78 (2–3), 223–236. http://dx.doi.org/10.1007/s11263-007-0105-5.

Chin, T., Cai, Z., Neumann, F., 2018. Robust fitting in computer vision: Easy or hard? In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds.), Computer Vision - ECCV 2018 - 15th European Conference. Munich, Germany, September 8-14, 2018, Proceedings, Part XII, In: Lecture Notes in Computer Science, vol. 11216, Springer, pp. 715–730. http://dx.doi.org/10.1007/978-3-030-01258-8_43.

Cohen-Steiner, D., Alliez, P., Desbrun, M., 2004. Variational shape approximation. vol. 23, (3), pp. 905–914. http://dx.doi.org/10.1145/1015706.1015817,

Edelsbrunner, H., Kirkpatrick, D.G., Seidel, R., 1983. On the shape of a set of points in the plane. IEEE Trans. Inf. Theory 29 (4), 551–558. http://dx.doi.org/10.1109/TIT.1983.1056714.

Fang, H., Lafarge, F., 2020. Connect-and-slice: An hybrid approach for reconstructing 3D objects. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR 2020, Seattle, WA, USA, June 13-19, 2020, Computer Vision Foundation / IEEE, pp. 13487–13495. http://dx.doi.org/10.1109/CVPR42600.2020.01350, URL https://openaccess.thecvf.com/content_CVPR_2020/html/Fang_Connect-and-Slice_An_Hybrid_Approach_for_Reconstructing_3D_Objects_CVPR_2020_paper.html.

Fang, H., Lafarge, F., Desbrun, M., 2018. Planar shape detection at structural scales. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, Computer Vision Foundation / IEEE Computer Society, pp. 2965–2973. http://dx.doi.org/10.1109/CVPR.2018.00313, URL http://openaccess.thecvf.com/content_cvpr_2018/html/Fang_Planar_Shape_Detection_CVPR_2018_paper.html.

Gao, W., Nan, L., Boom, B., Ledoux, H., 2021. SUM: a benchmark dataset of semantic urban meshes. CoRR abs/2103.00355, arXiv:2103.00355.

Garland, M., Heckbert, P.S., 1997. Surface simplification using quadric error metrics. In: Owen, G.S., Whitted, T., Mones-Hattal, B. (Eds.), Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH 1997, Los Angeles, CA, USA, August 3-8,1997, ACM, pp. 209–216. http://dx.doi.org/10.1145/258734.258849.

Guthe, M., Borodin, P., Klein, R., 2005. Fast and accurate hausdorff distance calculation between meshes. J. WSCG 13 (2), 41–48, URL http://wscg.zcu.cz/wscg2005/Papers_2005/Journal/!WSCG2005_Journal_Final.pdf.

Lafarge, F., Mallet, C., 2012. Creating large-scale city models from 3D-point clouds: A robust approach with hybrid representation. Int. J. Comput. Vis. 99 (1), 69–85. http://dx.doi.org/10.1007/s11263-012-0517-8.

Li, L., Sung, M., Dubrovina, A., Yi, L., Guibas, L.J., 2019. Supervised fitting of geometric primitives to 3D point clouds. In: IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, Computer Vision Foundation / IEEE, pp. 2652–2660. http://dx.doi.org/10.1109/CVPR.2019.00276, URL http://openaccess.thecvf.com/content_CVPR_2019/html/Li_Supervised_Fitting_of_Geometric_Primitives_to_3D_Point_Clouds_CVPR_2019_paper.html.

Li, M., Wonka, P., Nan, L., 2016. Manhattan-world urban reconstruction from point clouds. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.), Computer Vision - ECCV 2016 - 14th European Conference. Amsterdam, the Netherlands, October 11-14, 2016, Proceedings, Part IV, In: Lecture Notes in Computer Science, vol.9908, Springer, pp. 54–69. http://dx.doi.org/10.1007/978-3-319-46493-0_4.

Luo, Y., Mi, Z., Tao, W., 2021. DeepDT: Learning geometry from delaunay triangulation for surface reconstruction. In: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, the Eleventh Symposium on Educational Advances in Artificial Intelligence. EAAI 2021, Virtual Event, February 2-9, 2021, AAAI Press, pp. 2277–2285, URL https://ojs.aaai.org/index.php/AAAI/article/view/16327.

Marshall, A.D., Lukács, G., Martin, R.R., 2001. Robust segmentation of primitives from range data in the presence of geometric degeneracy. IEEE Trans. Pattern Anal. Mach. Intell. 23 (3), 304–314. http://dx.doi.org/10.1109/34.910883.

Mi, Z., Luo, Y., Tao, W., 2020. SSRNet: Scalable 3D surface reconstruction network. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR 2020, Seattle, WA, USA, June 13-19, 2020, Computer Vision Foundation / IEEE, pp. 967–976. http://dx.doi.org/10.1109/CVPR42600.2020.00105, URL https://openaccess.thecvf.com/content_CVPR_2020/html/Mi_SSRNet_Scalable_3D_Surface_Reconstruction_Network_CVPR_2020_paper.html.

Nan, L., Wonka, P., 2017. PolyFit: Polygonal surface reconstruction from point clouds. In: IEEE International Conference on Computer Vision. ICCV 2017, Venice, Italy, October 22-29, 2017, IEEE Computer Society, pp. 2372–2380. http://dx.doi.org/10.1109/ICCV.2017.258.

Oesau, S., Lafarge, F., Alliez, P., 2014. Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut. ISPRS J. Photogramm. Remote Sens. 90, 68–82.

Papadimitriou, C.H., Steiglitz, K., 1982. Combinatorial Optimization: Algorithms and Complexity. Prentice-Hall.

Rabbani, T., Van Den Heuvel, F., Vosselmann, G., 2006. Segmentation of point clouds using smoothness constraint. Int. Arch. Photogram. Remote Sensing Spatial Inf. Sci. 36 (5), 248–253.

Salinas, D., Lafarge, F., Alliez, P., 2015. Structure-aware mesh decimation. 34, (6), pp. 211–227. http://dx.doi.org/10.1111/cgf.12531,

Schindler, F., Förstner, W., Frahm, J., 2011. Classification and reconstruction of surfaces from point clouds of man-made objects. In: IEEE International Conference on Computer Vision Workshops. ICCV 2011 Workshops, Barcelona, Spain, November 6-13, 2011, IEEE Computer Society, pp. 257–263. http://dx.doi.org/10.1109/ICCVW.2011.6130251.

Schnabel, R., Wahl, R., Klein, R., 2007. Efficient RANSAC for point-cloud shape detection. Comput. Graph. Forum 26 (2), 214–226. http://dx.doi.org/10.1111/j.1467-8659.2007.01016.x.

van Kreveld, M.J., van Lankveld, T., Veltkamp, R.C., 2011. On the shape of a set of points and lines in the plane. Comput. Graph. Forum 30 (5), 1553–1562. http://dx.doi.org/10.1111/j.1467-8659.2011.02029.x.

Verdie, Y., Lafarge, F., Alliez, P., 2015. LOD generation for Urban scenes. ACM Trans. Graph. 34 (3), 30:1–30:14. http://dx.doi.org/10.1145/2732527.

Vigerske, S., Gleixner, A.M., 2018. SCIP: global optimization of mixed-integer nonlinear programs in a branch-and-cut framework. Optim. Methods Softw. 33 (3), 563–593. http://dx.doi.org/10.1080/10556788.2017.1335312.

Williams, H.P., 2009. Logic and integer programming. In: Logic and Integer Programming, vol. 130, Springer, http://dx.doi.org/10.1007/978-0-387-92280-5.